Jurnal Teknologi

Full paper

A Comparison of Second Order and Non-Second Order Limit Language Generated by Yusof-Goode Splicing System

Muhammad Azrin Ahmad[a], Nor Haniza Sarmin[a,*], Wan Heng Fong[b], Yuhani Yusof[c]

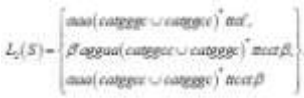
[a]*Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor Malaysia*
[b]*Ibnu Sina Institute for Fundamental Science Studies, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor Malaysia*
[c]*Faculty of Industrial Science and Technology, Universiti Malaysia Pahang, 26300 UMP Gambang, Pahang*

*Corresponding author: nhs@utm.my

Article history

Received: 8 June 2014
Received in revised form:
4 October 2014
Accepted: 1 December 2014

Graphical abstract



Abstract

DNA splicing process is a study on the recombinant behavior of double-stranded DNA molecules with the existence of restriction enzyme and ligase. Head introduced the first mathematical model of splicing systems by using the relation of informational macromolecules and formal language theory. In addition, a few laboratory experiments have been conducted in order to verify certain types of splicing language called inert/adult, transient and limit language. Previously, researchers have focused on those types of splicing languages. Recently, an extension of limit languages namely second order limit language has been introduced. In this paper, the difference between second order limit languages and non-second order limit languages is depicted in some examples. Then, the formations of second order limit language in Yusof-Goode splicing system are investigated.

Keywords: Y-G splicing system; Y-G splicing language; second order limit language

Abstrak

Proses hiris-cantum DNA ialah satu kajian tingkah laku rekombinan molekul-molekul DNA bebenang dua dengan kehadiran enzim pembatas dan ligase. Head telah memperkenalkan permodelan matematik pertama bagi sistem hiris-cantum dengan menggunakan hubungan makromolekul berinformasi dan teori bahasa formal. Tambahan pula, beberapa eksperimen makmal telah dijalankan untuk mengesahkan beberapa jenis bahasa hiris-cantum iaitu lengai/dewasa, fana dan bahasa batas. Sebelum ini, penyelidik-penyelidik telah memfokuskan jenis-jenis bahasa hiris-cantum tersebut. Kebelakangan ini, lanjutan kepada bahasa batas iaitu bahasa batas berperingkat dua telah diperkenalkan. Dalam kertas kerja ini, perbezaan bahasa batas berperingkat dua dan bahasa batas tidak berperingkat dua digambarkan melalui beberapa contoh. Kemudian pembentukan bahasa batas berperingkat dua dalam sistem hiris-cantum Yusof-Goode dikaji.

Kata kunci: Sistem Y-G hiris-cantum; bahasa Y-G hiris-cantum; bahasa batas berperingkat dua

1.0 INTRODUCTION

Deoxyribonucleic acid (DNA) plays an important role in a living organism. DNA is made up of three distinct components, which are sugar, phosphate group and nitrogenous base[1]. A nitrogenous base consists of four bases; adenine (*A*), guanine (*G*), cytosine (*C*) and thymine (*T*). The bases can be grouped into two; pyrimidines (*C* and *T*) and purines (*A* and *G*). Based on Watson-Crick complementarity, the only possible pairings are *A* with *T* and *C* with *G*, and vice versa (refer to Figure 1[1].

Restriction enzyme has a significant role on DNA molecules. It can cut the DNA molecules within the identified region called recognition sites[2]. When the restriction enzyme acts on the DNA molecules, two fragments with complementary sticky ends or blunt ends will exist. The pasting process then takes place with

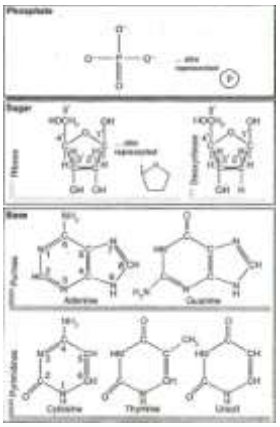


Figure 1 The components of DNA molecule

the presence of ligase when two conditions are met: the two fragments end with complementary bases, and the two fragments are from the same overhang. Then, either a new hybrid DNA or the same DNA molecule is formed.

Head, who is the pioneer in the mathematical modelling of the recombinant behavior of DNA molecules introduced splicing systems first that links formal language theory to the study of informational macromolecules[3]. The components of a splicing system consist of an alphabet A, a set of initial strings or axioms I, and a set of rules R [4].

The complementary bases of DNA can be represented in the form of [A/T], [C/G], [G/C], [T/A] or simply a, c, g, t where the bases can be seen as a sequence over the four-alphabets[5]. The restriction enzyme represents the set of rules in a splicing system.

There are several splicing system models namely Paun, Pixton, Goode-Pixton and Yusof-Goode (Y-G) splicing systems[6]. As time evolves, it can be seen that the splicing system models can be categorised into two different categories: a model based on the generation of language, and a model to preserve the biological characteristics of splicing process[7]. In this paper, we focus on the preservation of biological characteristics of splicing process, where Y-G splicing system presents the transparent behavior of the DNA biological process.

Based on splicing systems, a collection of DNA molecules that is produced when the restriction enzyme reacts with the initial string is called the splicing language. They are a few types of splicing languages namely inert/adult, transient and limit languages. These types of splicing language can be verified by laboratory experiments. Firstly, Laun and Reddy[8] conducted an experiment to verify the existence of predicted splicing language using two enzymes namely *Bgl*I and *Dra*III. The experiment was conducted at one stage only where two enzymes were included at a time. In 2008, Fong[9] used two enzymes namely *Aci*I and *Hpa*II at two stages to validate the adult and limit language. Next, Yusof[6] also used two enzymes namely *Acl*I and *Aci*I to show the non-uniqueness of limit language. Yusof[6] in also renamed the inert language to inert persistent language and introduced a new definition of splicing language namely active persistent language that is a set of strings which is involved in further splicing and is also contained in the limit language. In the recent year, Karimi[7] conducted a laboratory experiment using another two enzymes namely *Acc*65I and *CviQ*I to validate the behaviour of persistent splicing systems. The first two experiments were conducted with the presence of each enzyme separately. In the next experiment, two enzymes were added simultaneously. The splicing languages obtained from the three experiments are the same.

According to the experiment, a limit language is a type of splicing language which results as the remaining molecules after the splicing system has reached its equilibrium state or is completed. Consequently, limit language can be extended to second order limit language[10], where the second order limit language is deduced from the definition of n^{th} order limit language.

In this paper, the definition and the sufficient conditions for Y-G splicing system to produce second order limit language are given. The difference between second order limit language and non-second order limit language are illustrated through some examples. Then, the formations of second order limit language in the Y-G splicing system of at most two initial strings and two rules are discussed and presented as theorems.

2.0 PRELIMINARIES

In this section, some fundamental definitions of splicing systems and certain types of splicing languages which are used in determining the further results in this paper are given.

The first three definitions relate to formal language theory.

Definition 2.1[4]: Alphabet

An alphabet, A, is a finite, nonempty set of symbols.

Definition 2.2[4]: String

A string is a finite sequence of symbols from the alphabet.

Definition 2.3[4]: Language

A set of strings all of which are chosen from some A^*, where A is a particular alphabet, is called a language.

Note that, A^* denotes the set of all strings over an alphabet A which is obtained by concatenating zero or more symbols from A.

The definition of Y-G splicing system that will be used throughout this paper is presented next.

Definition 2.4[6]: Y-G Splicing System

A splicing system $S=(A,I,R)$ consists of a set of alphabets A, a set of initial strings I in A^* and a set of rules, $r \in R$ where $r = (u, x, v; y, x, z)$. For $s_1 = \alpha uxv\beta$ and $s_2 = \gamma yxz\delta$ elements of I, splicing s_1 and s_2 using r produces the initial string I together with $\alpha uxz\delta$ and $\gamma yxv\beta$, presented in either order where $\alpha, \beta, \gamma, \delta, u, x, v, y$ and $z \in A^*$ are the free monoid generated by A with the concatenation operation and 1 as the identity element.

Two types of splicing languages are discussed in this paper, namely *transient* and limit languages. Experimentally, a splicing language is called *transient* if a set of strings is eventually used up and disappear in a given system. Other than that, a splicing language is a limit language given that it is the set of words that are predicted to appear if some amount of each initial molecule is present, and sufficient time has passed for the reaction to reach its equilibrium state, regardless of the balance of the reactants in a particular experimental run of the reaction.

In the following, the definition of n^{th} order limit language is given followed by the definition of second order limit language.

Definition 2.5[10]: n^{th} Order Limit Language

Let the set L_n of n^{th} order limit words of L to be the set of first order limit word of L_{n-1}. We obtain L_n from L_{n-1} by deleting the words that are transient in L_{n-1}.

Consequently when $n = 2$, the following definition holds.

Definition 2.6[11]: Second Order Limit Language

Let the set L_2 of second order limit words of L to be the set of first order limit words of L_1. We obtain L_2 from L_1 by deleting words that are transient in L_1.

3.0 RESULTS AND DISCUSSION

In this section, three examples are illustrated to show the difference between second order limit languages and non-second order limit languages.

3.1 Molecular Examples of Non-Second Order Limit Language and Second Order Limit Language

Two actual examples of non-second order limit language and an example of second order limit language are given. Experimentally, a test tube containing DNA template are chosen from enterobacteria phage lambda digested with *Hind*III from New England Biolabs. Besides that, the restriction enzymes[12] from are supplied together with the suitable buffer for robust production. By adding an appropriate ligase, the new molecules will be formed. The first two examples depicted non-second order limit languages.

In the following example, the restriction enzyme *Aci*I which is supplied together with CutSmart™ buffer is chosen in the splicing process below.

Example 3.1

Let $S=(A,I,R)$ be a Y-G splicing system consisting of a set of alphabets, $A=\{a,c,g,t\}$, a set of initial strings, $I=\{\alpha ccgc\beta\}$ such that α with α' and β with β' are complement to each other where $\alpha,\beta,\alpha',\beta'\in A^*$ and a set of rules, $R=\{r\}$ where $r=(c;cg,c:c;cg,c)$. In a solution, multiple copies of dsDNA molecules are present. The 180 degrees rotation of the initial string is also considered in the splicing process.

$$I_0=\begin{matrix}5'-\alpha C & CGC\beta-3'\\ 3'-\alpha'GGC & G\beta'-5'\end{matrix},\ I_{180}=\begin{matrix}5'-\beta'G & CGG\alpha'-3'\\ 3'-\beta CGC & C\alpha-5'\end{matrix}.$$

When first spliced, the following splicing language is generated:

$$(\alpha ccgc\beta)\xrightarrow{R}\{\alpha ccgc\beta,\alpha ccgg\alpha',\beta'gcgc\beta\}.$$

When the splicing occurs once again, no new string is formed. Hence, this Y-G splicing system does not produce second order limit language.

In the next example, there are two DNA sequences and two restriction enzymes in the Y-G splicing system. The example is conducted at one stage only where the restriction enzymes are added simultaneously to the splicing process. The restriction enzymes namely *Aci*I and *HinP*1I and a suitable buffer that works for both restriction enzymes namely CutSmart™ is added to the reaction for robust production.

Example 3.2

Let $S=(A,I,R)$ be a Y-G splicing system consisting of a set of alphabets, $A=\{a,c,g,t\}$, a set of initial strings, $I=\{\alpha ccgc\beta,\gamma gcgc\delta\}$ such that α with β, γ with δ, α' with β', γ' with δ' are complement to each other where $\alpha,\beta,\gamma,\delta,\alpha',\beta',\gamma',\delta'\in A^*$ and a set of rules, $R=\{r_1,r_2\}$ where $r_1=(c;cg,c:c;cg,c)$ and $r_2=(g;cg,c:g;cg,c)$. When splicing occurs, the following splicing language is generated:

$$\{\alpha ccgc\beta,\gamma gcgc\delta\}\xrightarrow{R}\begin{Bmatrix}\alpha ccgc\beta,\alpha ccgg\alpha',\beta'gcgc\beta,\\ \gamma gcgc\delta,\gamma gcgc\gamma',\delta'gcgc\delta,\\ \alpha ccgc\delta,\gamma gcgc\beta,\alpha ccgc\gamma',\delta'gcgc\beta\end{Bmatrix}.$$

Again, this Y-G splicing system does not produce second order limit language since no new string is formed when the splicing process occurs once again.

The following example is given to show the presence of second order limit language in the Y-G splicing system. The restriction enzyme *Fat*I which is supplied together with NEBuffer 2.1 is chosen in the splicing process below.

Example 3.3

Let $S=(A,I,R)$ be a Y-G splicing system consisting of a set of alphabets, $A=\{a,c,g,t\}$, a set of initial strings, $I=\{\alpha aacatgggccatgttcct\beta\}$ such that α with α' and β with β' are complement to each other where $\alpha,\beta,\alpha',\beta'\in A^*$ and a set of rules, $R=\{r\}$ where $r=(1;catg,1:1;catg,1)$. When splicing occurs, the following splicing language is generated:

$$\{\alpha aacatgggccatgttcct\beta\}\xrightarrow{R}I\cup\begin{Bmatrix}\alpha aacatgttcct\beta,\alpha aacatgtt\alpha',\\ \beta'aggaacatgttcct\beta,\\ \alpha aacatgggccatgtt\alpha',\\ \alpha aacatggcccatgttcct\beta,\\ \beta'aggaacatgggccatgttcct\beta,\\ \alpha aacatgggccatggcccatgttcct\beta,\\ \alpha aacatgggccatggcccatgtt\alpha',\\ \alpha aacatggcccatgggccatgttcct\beta,\\ \alpha aacatggcccatgggccatgtt\alpha',\\ \beta'aggaacatgggccatggcccatgttcct\beta,\\ \beta'aggaac\arg ggccatgggccatgttcct\beta\end{Bmatrix}.$$

When the splicing process occurs again, the following second order limit language is obtained as follows:

$$L_2(S)=\begin{Bmatrix}\alpha aa(catgggc\cup catggcc)^*catgtt\alpha',\\ \beta'aggaa(catggcc\cup catgggc)^*catgttcct\beta,\\ \alpha aa(catggcc\cup catgggc)^*catgttcct\beta\end{Bmatrix}.$$

In formal language theory, the symbol (*) on string $\alpha aa(catgggc\cup catggcc)^*catgtt\alpha'$ indicates that either *catgggc* or *catggcc* can occur alternately. It also shows that the number of occurrence or repetition of string *catgggc* and *catggcc* is independent such that for an example, the string *catgggc* occurs only once while the string *catggcc* occurs twice and so on. Hence, it is clearly shown that the second order limit language exists.

3.2 The Formation of Second Order Limit Language

The existence of second order limit language in few cases are discussed and presented in the following theorems.

Theorem 3.1

Let $S=(A,I,R)$ be a Y-G splicing system where A is a set of alphabets, $I=\{\alpha abcd\beta\}$ is a set of initial strings such that $\alpha,\beta,\alpha',\beta'\in A^*$ and $R=\{r_1,r_2\}$ is a set of rules where $r_1=(a;bc,d:a;bc,d)$ and $r_2=(a;bc,a':a;bc,a')$ such that a with a', b with b', c with c' and d with d' are complement to each other. If the splicing process is conducted in two stages, then the second order limit language exists.

Proof

From the given splicing system $S=(A,I,R)$, the following splicing language is produced:

$$\{\alpha abcd\beta\}\xrightarrow{r_1} I\cup\{\alpha abca'\alpha',\beta'd'bcd\beta\}.$$

In the next splicing process, the second rule is applied only to the string with its recognition site. The resulted splicing language is stated as below:

$$\{\alpha abca'\alpha'\}\xrightarrow{r_2}\{\alpha abca'\alpha'\}.$$

Therefore, there exists second order limit language, $L_2(S)=\{\alpha abca'\alpha'\}$. Hence, the proof is complete.□

Theorem 3.2

Let $S=(A,I,R)$ be a Y-G splicing system where A is a set of alphabets, $I=\{\alpha aabbcdcd\beta\}$ is a set of initial strings and the crossing sites of the rules are disjoint, $R=\{r_1,r_2\}$ where $r_1=(a;ab,b:a;ab,b)$ and $r_2=(c;dc,d:c;dc,d)$ such that $ab\neq cd$. Then the second order limit language exists.

Proof

From the given splicing system $S=(A,I,R)$, the following splicing language is produced:

$$\{\alpha aabbcdcd\beta\}\xrightarrow{R} I\begin{Bmatrix}\alpha aabb\alpha',\beta'cdcd\beta,\\ \alpha aabbcdcdaabb\alpha',\beta'cdcdaabbcdcd\beta\end{Bmatrix}.$$

When the splicing process occurs again, the second order limit language is obtained as follows:

$$L_2(S)=\begin{Bmatrix}\alpha aabbc(dcda\cup abbc)^j dcd\beta,\\ \alpha a(abbc\cup dcda)^j abb\alpha',\\ \beta'c(dcda\cup abbc)^j dcd\beta\end{Bmatrix}\text{ for } j=0,1,\cdots,n.$$

Hence, the second order limit language exists. □

Theorem 3.3

Let $S=(A,I,R)$ be a Y-G splicing system where A is a set of alphabets, $I=\{\alpha axbxc\beta\}$ is a set of initial string with two cutting sites and $R=(1;x,1:1;x,1)$ such that $x\in A^*$ is a set of rules given that α with β, a with b and also c with d is complement to each other, $\alpha,\beta,a,b,c,d\in A^*$, then the second order limit language exists.

Proof

From the given splicing system $S=(A,I,R)$, the following splicing language is produced:

$$\{\alpha axbxc\beta\}\xrightarrow{R} I\cup\begin{Bmatrix}\alpha axc\beta,\alpha axb\alpha',\beta'dxc\beta,\alpha axbxb\alpha',\\ \alpha axaxc\beta,\beta'dxbxc\beta,\alpha axbxaxc\beta,\\ \alpha axbxaxb\alpha',\alpha axaxbxc\beta,\\ \alpha axaxbxb\alpha',\beta'dxbxaxc\beta,\beta'xaxbxc\beta\end{Bmatrix}.$$

When the splicing process occurs again, the second order limit language is obtained as follows:

$$L_2(S)=\{\alpha a(xb\cup xa)^* xc\beta,\beta'd(xa\cup xb)^* xb\beta,\alpha a(xa\cup xb)^* xb\alpha'\}.$$

Hence, the second order limit language exists. □

4.0 CONCLUSION

In this paper, Example 3.1 to Example 3.3 show the difference between second order limit language and non-second order limit language in biomolecular context. Some theorems on the formation of second order limit language are given in Theorem 3.1 to Theorem 3.3. Hence it can be concluded that the second order limit language exists when the splicing system is conducted at two stages, crossing sites of the set of rules are disjoint and the set of initial strings has two cutting sites.

Acknowledgement

The first author is indebted to the Ministry of Education (MOE) Malaysia for his financial funding through MyBrain15 (MyPhD) scholarship. The second and third authors would like to thank MOE and Research Management Centre (RMC), Universiti Teknologi Malaysia (UTM) for the financial funding through UTM Research University Grant Vote No. 08H45.

References

[1] Alcamo, I.E. 2001. 2nd Edition. *DNA Technology Awesome Skill*. USA: Academic Press.
[2] Tamarin, R.H. 2001. 7th Edition. *Principle of Genetics*. USA: The Mac-Graw Hill Companies.
[3] Head, T. 1987. Formal Language Theory and DNA: An Analysis of the Generative Capacity of Specific Recombinant Behaviors. *Bulletin of Mathematical Biology*. 49(6): 737–759.
[4] Linz, P. 2006. 4th Edition. *An Introduction to Formal Languages and Automata*. USA: Jones and Barlett Publishers, Inc.
[5] Paun, G., G. Rozenberg and A. Salomaa. 1998. *DNA Computing New Paradigms*. Germany: Springer-Verlag.

[6] Yusof, Y. 2012. *DNA Splicing System Inspired by Bio Molecular Operations*. Ph.D. Thesis, Universiti Teknologi Malaysia.
[7] Karimi, F. 2013. *Mathematical Modelling of Persistent Splicing Systems in DNA Computing*. Ph.D Thesis. Universiti Teknologi Malaysia.
[8] Laun, E. and K. J. Reddy. 1997. Wet Splicing Systems. DIMACS Series in *Discrete Mathematics and Theoretical Computer Science*. 48: 73–83.
[9] Fong, W.H. 2008. *Modelling of Splicing Systems Using Formal Language Theory*. Ph.D Thesis. Universiti Teknologi Malaysia.
[10] Goode, E. and D. Pixton. 2004. Splicing to the Limit. *Lecture Notes in Computer Science*. 2950: 189–201.
[11] Ahmad, M.A., N.H. Sarmin, W.H. Fong and Y. Yusof. 2014. An Extension of First Order Limit Language. *AIP Proceeding*. 1602: 627–631.
[12] Research Biolabs Sdn. Bhd. 2011. *New England Biolabs 2011-12 Catalogue and Technical Reference*. USA: Catalogue.